Protected Sharing of 3D Models of Cultural Heritage and Archaeological Artifacts

David KOLLER

University of Virginia, Institute for Advanced Technology in the Humanities
koller@virginia.edu

Abstract

The increasing use of three-dimensional modeling and digitization techniques in archaeology has raised questions about the digital rights management of the resulting 3D models. Curators of valuable artifacts and creators of 3D models may be reluctant to openly share the digital 3D representations due to fear of misuse or theft of their data and intellectual property. In order to address these concerns, we have investigated a number of techniques for sharing and disseminating 3D models in a secure, protected manner. We have implemented and deployed one such technique, remote 3D rendering, on a wide scale. Our remote rendering system has been used successfully to make thousands of 3D models freely available for interactive visualization, including archaeological artifacts, digitized statuary, and reconstructions of ancient urban sites, while preventing copying or other unauthorized access to the underlying 3D data.

Keywords

3D models, cultural heritage, digital rights management, remote rendering

1. Introduction

The increasing use of three-dimensional modeling and digitization techniques in archaeology has raised questions about the digital rights management of the resulting 3D models. While the digital rights management problem of protecting data from theft and misuse has previously been addressed for a variety of other information types (software code, digital 2D images, audio and video files), few technological solutions are designed specifically to protect interactive 3D graphics content.

The demand for protecting 3D graphical models is significant and growing, driven by needs in the cultural heritage community. Contemporary 3D digitization technologies allow the efficient creation of accurate 3D models of many physical objects. The Stanford Digital Michelangelo Project, for example, has developed a high-resolution digital archive of 10 of Michelangelo's large statues, including the David (Levoy *et al.* 2000). These statues represent the artistic patrimony of Italy's cultural institutions, and the contract with the Italian authorities permits the distribution of the 3D models only to established scholars for non-commercial use. Although everyone involved would like the models to be widely available for all constructive purposes, the digital 3D models of the statues might be pirated and misused if they were distributed without protection. For example, simulated marble replica statuettes of the David might be manufactured from the 3D data, violating the provisions of the contracts that authorized the digitization and creation of the model.

Digital archives of archaeological artifacts are another example of cultural heritage 3D models that might require digital rights protection. Curators of such artifact collections increasingly turn to 3D digitization as a way to preserve and widen scholarly use of their holdings, but they often desire to maintain strict control over the use of the 3D data and guard against theft. An example of such a collection is the Digital Forma Urbis Romae Project (Koller *et al.* 2006), a collaboration with Italian archaeological officials that has digitized over one thousand marble fragments of an ancient Roman map and made them publically available through a web-based database, while taking measures to prevent copying of the 3D data. Other 3D graphics application areas with intellectual property concerns include character modeling for animated films, content for video games, human body scans, CAD, and online commerce (VRML, etc.).

Prior technical research in the area of intellectual property protection for 3D data has primarily addressed 3D digital watermarking techniques. These steganographic approaches have sought to embed hidden information into 3D graphical models, with varying degrees of robustness to attacks aimed at disabling the watermarks by altering the 3D shape or data representation. Many of the most successful 3D

watermarking schemes are based on spread-spectrum frequency domain transformations, embedding watermarks at multiple scales by introducing controlled perturbations into the coordinates of the 3D model vertices (Praun *et al.* 1999). Complementary technologies search collections of 3D models and examine them for the presence of digital watermarks, in an effort to detect piracy.

To adequately protect valuable 3D objects such as cultural heritage artifacts, however, it is not sufficient to detect piracy after the fact; we must instead prevent it from happening in the first place. The computing industry has experimented with a number of techniques for preventing unauthorized access to digital data, including physical dongles, node-locked and networked licensing schemes, copy prevention software, obfuscation, and encryption with embedded keys. Most are these schemes can eventually be broken or bypassed by determined attackers, causing undue inconvenience and expense for nonmalicious users. High-profile data and software is particularly susceptible to attackers.

Fortunately, 3D graphics data differs from most other forms of digital media in that the presentation format, 2D images, is fundamentally differently from its underlying representation as 3D geometry. Usually, 3D graphics data is visualized as a projection onto a 2D display device, resulting in a large information loss for single views. This property suggests that protected 3D graphics systems can perhaps still be highly useful to users, without making all the 3D data as vulnerable to piracy as other types of digital content.

Our goal is to address the problem of preventing the theft of 3D cultural heritage models, while still sharing them and allowing for their interactive display and manipulation. We attempt to provide a solution for maintainers of large collections of high-resolution static 3D models (such as digitized cultural heritage artifacts). The methods we have developed aim to protect both the physical shape of the 3D models and their particular geometric representation (such as 3D mesh vertex coordinates, surface normals, and connectivity information). We accept that the coarse shape of visible objects can be easily reproduced regardless of any protection efforts, so we concentrate on defending the high-resolution geometric detail of 3D models. This detailed geometry is usually the most expensive to model or measure (perhaps requiring special access and advanced 3D digitizing technology), and is often the most valuable in exhibiting fidelity to the original object.

2. Protection Techniques for 3D Models

Data in the 3D graphics pipeline is vulnerable to an attacker in a variety of ways when displaying a 3D model on a personal computer. The possible means of attack include:

- 3D model file reverse-engineering. If users have full access to 3D model data files, they can reverse-engineer even obfuscated or encrypted file formats.
- 3D viewer application tampering. Hackers can use techniques such as program tracing and memory dumping to obtain access to data in use by application programs.
- Graphics driver tampering. 3D data passes through graphics driver software on its way to the graphics hardware; the drivers are vulnerable to tampering or replacement by attackers to capture streams of 3D data.
- Reconstruction from the framebuffer. Sophisticated attackers could access rendered images from the graphics memory and use 3D computer vision techniques to reconstruct the original model.
- Reconstruction from the final image display. Regardless of any system protections in the pipeline, the final video images output from a graphics system are also vulnerable to capture and reconstruction.

To counter these possible attacks, we have considered several possible approaches for protected rendering of 3D graphics:

- Software-only rendering. By bypassing the graphics processing unit (GPU) driver and hardware, a strict software rendering approach can maintain complete control of the rendering process within a specialized 3D viewing application that uses obfuscation techniques to protect the 3D data in the early stages of the pipeline, at the expense of trading off display performance.
- Hybrid hardware/software rendering. To partially leverage GPU acceleration, one can render a subset of the model via software, and the rest via hardware. Alternatively, the transform and lighting stage of rendering can be performed in

software, while leaving the rasterization stage to the graphics hardware.

- Deformations in the geometry. The 3D viewing application can introduce subtle deformations in the geometry of the 3D model before passing the 3D vertex data to the graphics driver, so that attackers would have difficulty reconstructing the full 3D model due to the distortions.
- Hardware decryption in the GPU. If 3D models were encrypted using public-key encryption when they are created, then specially-designed GPUs could accept this encrypted data and perform on-chip decryption and rendering.
- Image-based rendering. Image-based graphics data representations, such as light fields (Levoy and Hanrahan 1996), are densely sampled data structures that do not explicitly include a geometric description for the 3D shape, yet are still amenable to interactive and accurate display.
- Encrypted computation rendering. Recent research in encrypted computation suggests that 3D rendering directly from encrypted representations of the 3D model may be possible, although the computational complexity is extremely high and currently prohibitive.
- Remote (network) rendering. The 3D model data can be retained on a secure server, under the contol of the content owner, and the server pass only 2D rendered images of the models back to user client requests. The 3D geometry is thus safe from all types of graphics pipeline attacks (except reconstruction from images), although the server itself is still vulnerable to direct attack.

After experimenting with a number of these approaches, we have pursued remote rendering as the best solution for protected sharing of 3D cultural heritage models. Implementing many of the other methods relies on "security through obfuscation," which is unsound from a computer security perspective. Hardware GPU decryption is a robust idea, but will require industry standardization before it is widely deployable. Image-based techniques require huge unwieldy data files, and encrypted computation rendering methods need further fundamental research before they might be tractable.

3. Remote Rendering for Protected Sharing of 3D Models

We have developed a remote rendering system with a client-server architecture to provide controlled, protected access to collections of 3D cultural heritage models (*Fig. 1*). Users employ a special 3D client viewer program to interactively view the protected 3D content. The program includes very low-resolution, decimated versions of the 3D models, that can be interactively rotated, zoomed, and illuminated by the user in real time. When the user stops manipulating the low-resolution model, detected by a "mouse up" event, the client program queries the remote rendering server via the network for a matching image rendered from the high-resolution model data, which replaces the low-resolution rendering seen by the user (*Fig. 2*).

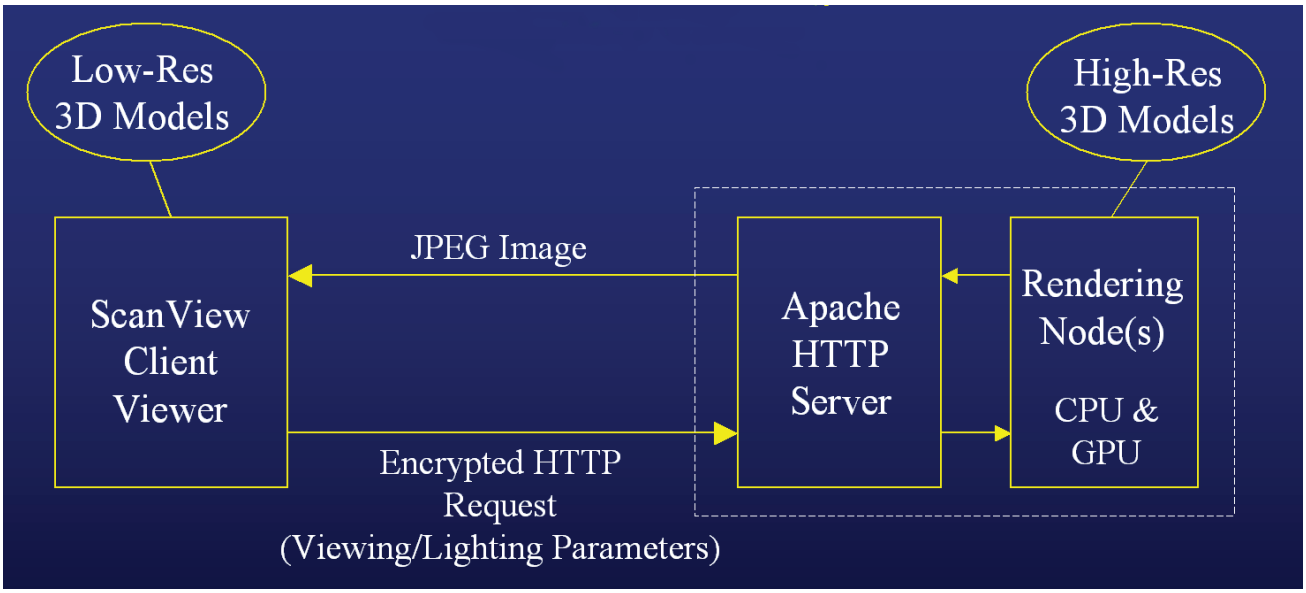

Fig. 1. Architecture of the remote rendering system.

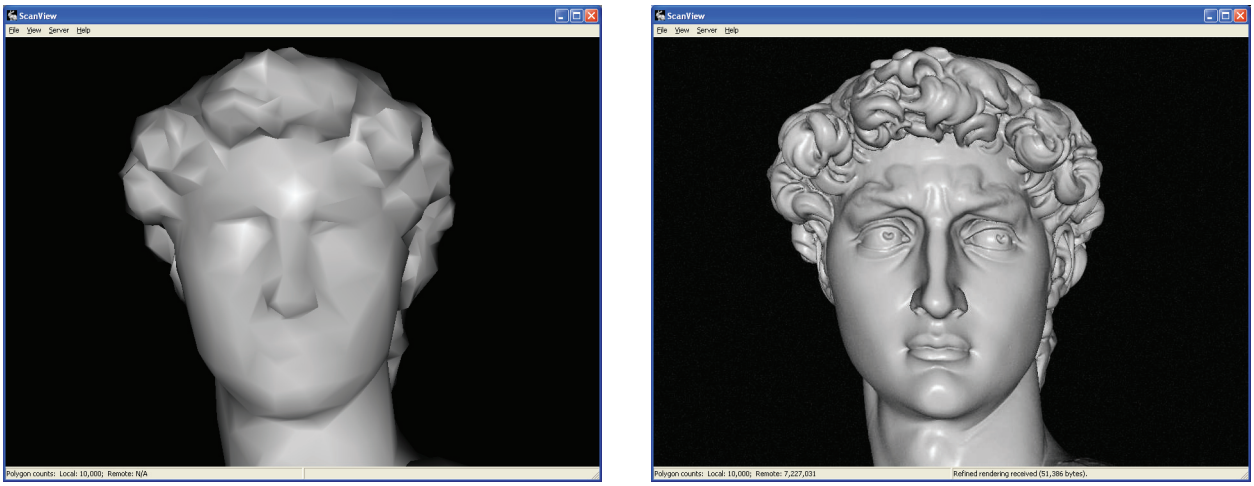

Fig. 2. Client-side low resolution (left) and server-side high resolution (right) model renderings.

On computer networks with reasonably low latencies, the user thus has the impression of manipulating a high-resolution version of the model. In typical use involving digitized cultural heritage artifacts, we use models with approximately 10,000 polygons for the low resolution version, whereas the server-side models often contain tens of millions polygons. Such low-resolution model complexities are of little value to potential attackers, yet still provide adequate reference for the user to navigate about the model.

The remote rendering server receives rendering requests from users' client programs, renders corresponding images, and passes them back to the clients. The rendering server is implemented as a module running under the Apache 2.0 HTTP Server, communicating with client programs using the standard HTTP protocol and taking advantage of the wide variety of access protection and monitoring tools built into the Web server software. As render requests are received from clients, the server checks their validity and dispatches valid requests to a GPU for OpenGL hardware-accelerated rendering. The rendered images are read back from the framebuffer, compressed using JPEG compression, and then returned to the client. The server uses level-of-detail techniques to speed the rendering of highly complex models and maintain high throughput rates. In practice, an individual server node with a Pentium 4 CPU and NVIDIA GeForce 4 video card can handle a maximum of 8 typical client requests per second; the bottlenecks are in the rendering and readback stage (about 100 milliseconds) and in the JPEG compression step (approximately 25 milliseconds). Incoming request sizes are about 700 bytes each, and the images returned from our servers average 30 kB per request.

The benefit of using a remote image rendering system to share 3D models is that the high-resolution model geometry data is never made available to potential attackers; only 3D reconstruction from the 2D images remains as a possible attack. General 3D reconstruction from images is a very challenging computer vision research problem. However, synthetic graphics renderings can be particularly susceptible to reconstruction, since the human effort to harvest a large number of images is low, and the attacker may be able to exactly specify the parameters used to create the images. Moreover, synthetic images are potentially perfect, with no sensor noise or miscalibration errors.

To combat such reconstruction attacks, we implement a number of defenses in our rendering server system. To deter image harvesting attacks, we perform automatic analysis of the server logs, detecting suspicious sequences or frequencies of image requests. We employ obfuscation to create hurdles for attackers by encrypting the rendering request messages sent from the client programs, as well as by encrypting the low-resolution client-side 3D models. The server imposes constraints on rendering requests, disallowing extremely close-up views of models, and requiring fixed view frustum dimensions. Finally, we add a number of perturbations and distortions to the images that are returned from the server. These image distortions are applied in a psuedorandom manner, so that their effects can not be easily modeled and reversed, and the magnitudes of the distortions are limited so as not to distract non-malicious users viewing the models. The types of distortions that we employ include nonlinear image warps, adding high-frequency noise to images,

and perturbing the lighting parameters slightly from those being requested by the client.

We have experimentally validated the effectiveness of these defenses against a variety of traditional computer vision reconstruction techniques, including shape-from-silhouette, shape-from-shading, and stereo reconstruction methods (Koller *et al.* 2004). Additionally, we have performed a series of psychological user studies to evaluate the perceptual effects of our distortions upon users of the graphics system, and to determine the ideal magnitudes for perturbations that minimize user distraction while still adequately defending against reconstruction attacks (Zhu *et al.* 2008). Ultimately, however, we know of no formalism for rigorously analyzing the security provided by our systems-based approach to protecting 3D models, and there is inevitably an "arms race" between the possible attacks and countermeasures.

4. Results

The protected graphics software client that we developed (named *ScanView*) has been made freely available for over four years. In that time, more than 50,000 unique users have downloaded the software and used it to access protected 3D archives containing approximately 1,300 different models. Many of these 3D models belong to prominent collections of laser scanned cultural heritage objects, including several statues of the Stanford Digital Michelangelo Project, and hundreds of archaeological fragments from the Digital Forma Urbis Romae Project. In addition, we have shared the server software with other institutions interested in hosting their 3D model archives using their own installation of the protected graphics system. We continue to seek collaborators in using these techniques, and believe that the results of our research is encouraging wider dissemination of digitized 3D cultural heritage objects that would otherwise be of limited access due to intellectual property concerns.

User feedback has been uniformly positive. Fetching high-resolution renderings over intercontinental broadband Internet connections incurs less than 2 seconds of latency, while faster continental connections generally experience latencies dominated by the processing time of the rendering server. The render server architecture can scale up to support an arbitrary number of requests per second, and servers can be installed at distributed locations around the world to reduce long distance latencies.

Users of our protected graphics systems have included archaeologists, sculptors, art students, and a wide variety of lay people. Few of them would have qualified under the strict guidelines required to obtain unrestricted access to the 3D models in the archives, so the protected remote rendering system has enabled whole new categories of users access to 3D graphical models for professional scholarship and personal enjoyment.

5. Future Work

We are actively continuing our development of tools for sharing 3D cultural heritage models in a protected manner. We are currently extending the ScanView client to have wider appeal and functionality. First, we are porting the client software to run inside Web browsers, rather than requiring the user to initially download a standalone executable program. Secondly, we are adding capability to the system to allow specification and remote rendering of complete video sequences. This approach uses a coarser-grained communication between the client and server, and may be appropriate when users desire to pre-compute a smooth walkthrough visualization of a large model such as an archaeological site.

One direction for further research is analysis of computer vision techniques that specifically address 3D reconstruction of synthetic data under antagonistic conditions, to increase our understanding of the efficacy of such attacks against the remote rendering server defenses. Another question is how to allow users a greater degree of geometric analysis of protected 3D models without further exposing the data to theft; scholarly users have expressed interest in measuring distances and plotting profiles of 3D objects for analytical purposes beyond the simple 3D viewing supported in the current remote rendering system. Finally, there is continued interest in alternative approaches to protecting 3D graphics besides remote rendering, including specialized systems that make data security a priority while sacrificing some general purpose computing platform capabilities. A GPU decryption scheme, for example, may be appropriate for console devices or other custom graphics systems.

References

Levoy, Marc, and Pat Hanrahan (1996). Light Field Rendering. In: *Proceedings of SIGGRAPH 1996*. ACM Press, 31–42.

Levoy, Marc, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk (2000). The Digital Michelangelo Project: 3D Scanning of Large Statues. In: *Proceedings of SIGGRAPH 2000*. ACM Press, 131–144.

Koller, David, Michael Turitzin, Marc Levoy, Marco Tarini, Giuseppe Croccia, Paolo Cignoni, and Roberto Scopigno (2004). Protected Interactive 3D Graphics Via Remote Rendering. *ACM Transactions on Graphics* 23, 3, 695–703.

Koller, David, Jennifer Trimble, Tina Najbjerg, Natasha Gelfand, and Marc Levoy (2006). Fragments of the City: Stanford's Digital Forma Urbis Romae Project. In: *Proceedings of the Third Williams Symposium on Classical Architecture, Journal of Roman Archaeology Suppl.* 61, 237–252.

Praun, Emil, Hughes Hoppe, and Adam Finkelstein (1999). Robust Mesh Watermarking. In: *Proceedings of SIGGRAPH 1999*. ACM Press, 49–56.

Zhu, Jiajun, Jonathan Bakdash, David Koller, Thomas Banton, Dennis Proffitt, and Greg Humphreys (2008). Quantifying Usability in Secure Graphics: Assessing the User Costs of Protecting 3D Content. In: *Proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV'08)*, ACM Press.